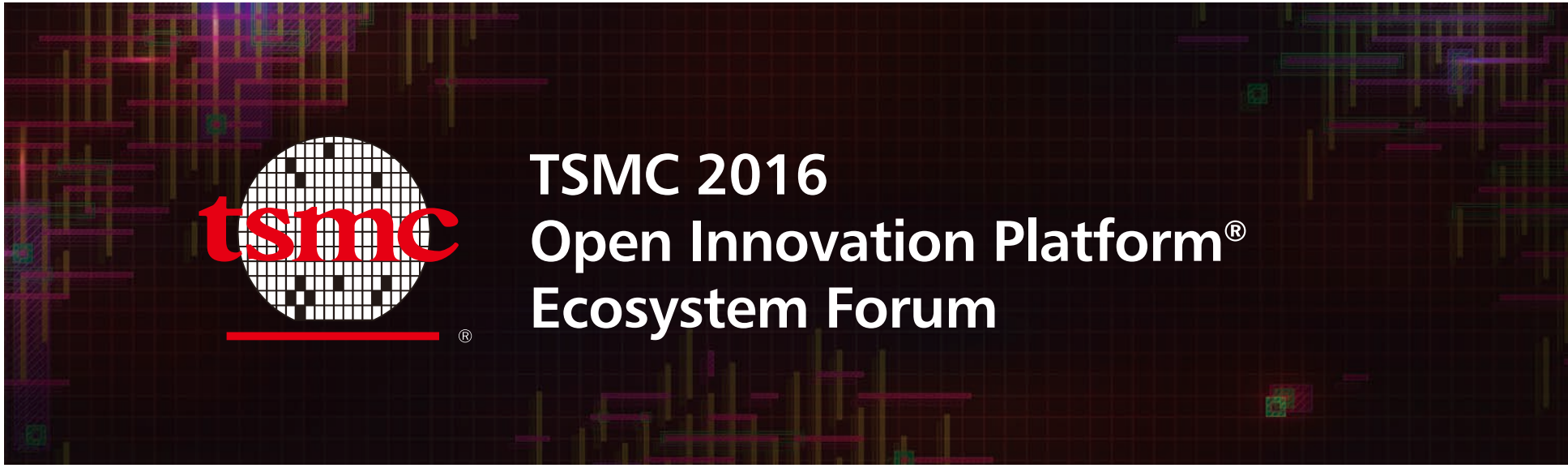
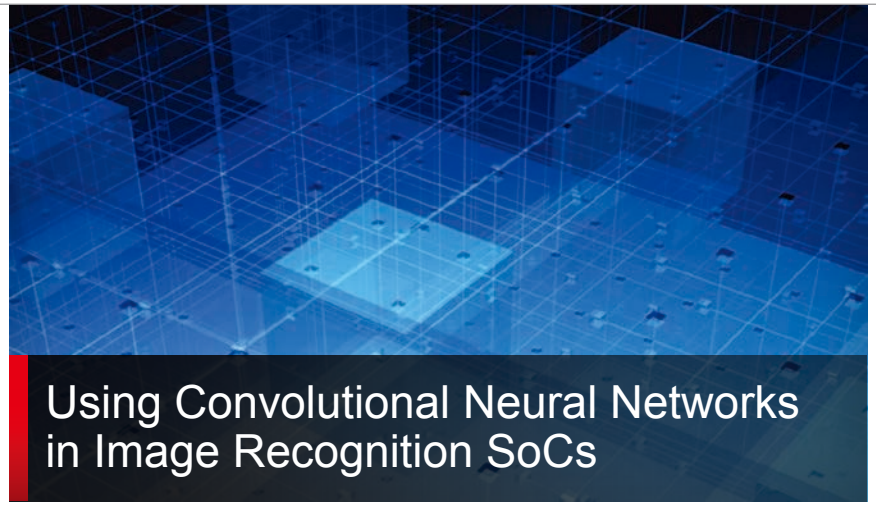
Using Convolutional Neural Networks in Image Recognition SoCs

Cadence



ABSTRACT

Convolutional neural networks (CNNs) became widely used in pattern- and image-recognition problems as they have a number of advantages compared to other techniques. This white paper covers the basics of CNNs including a description of the various layers used. Using traffic sign recognition as an example, we discuss the challenges of the general problem and introduce algorithms and implementation software developed by Cadence that can trade off computational burden and energy for a modest degradation in sign recognition rates. We outline the challenges of using CNNs in embedded systems and introduce the key characteristics of the Cadence Tensilica Vision P5 digital signal processor (DSP) for Imaging and Computer Vision and software that make it so suitable for CNN applications across many imaging and related recognition tasks. Also we highlight how is this new revolution in technology will drive the demand for large SoC more than ever before.



Using Convolutional Neural Networks in Image Recognition SoCs

Samer Hijazi, Sr. Design Engineering Architect
TSMC OIP Ecosystem Forum
San Jose, California
September 22, 2016

cadence

Outline

Cadence's mission
- Enable ***better***, ***faster***, ***cooler*** silicon systems ***sooner***

Imaging/video recognition
- Strong driver for creating advanced SoCs

Neural networks are a crucial innovation
- But, need a breakthrough in ***efficiency*** and **ease of development** for embedded use

  cadence

The Deep Learning Buzz

Speech recognition: Apple, Google, Nuance, Microsoft

Vision/ADAS: NVIDIA, Mobileye

Finance: TradeTrek, M.J. Futures, Alyuda

Social media and big-data search: Google, Facebook, Microsoft, Baidu, NEC, IBM, Yahoo, AT&T

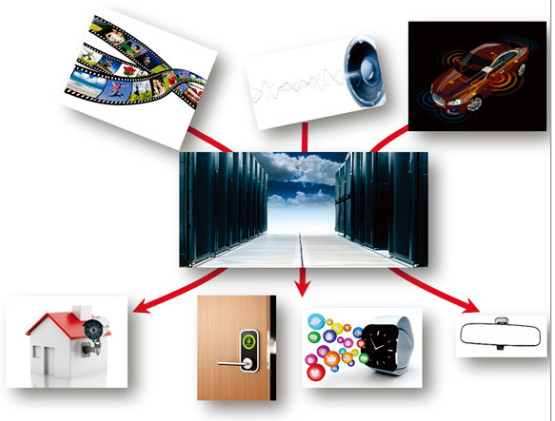
Medical: Genomics, radiology, screening, protein sequencing

Security: Google

  cadence

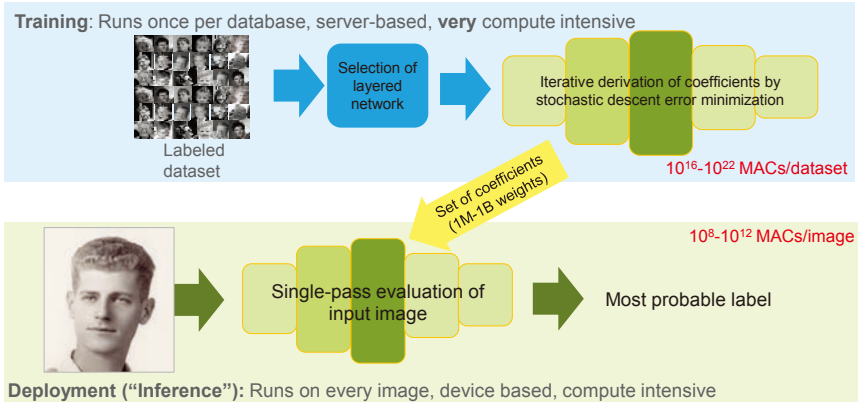
What Is the Embedded Electronics World Looking For?

- Create compelling embedded, real-time applications
- Transition real-time neural networks from the server world to the embedded world
- Develop software and hardware tools that enable practical solutions



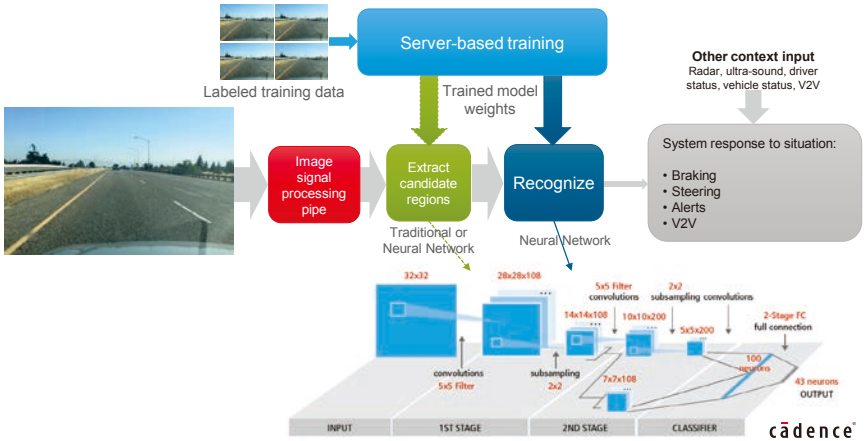
  cadence

The Basics of Real-Time Neural Networks



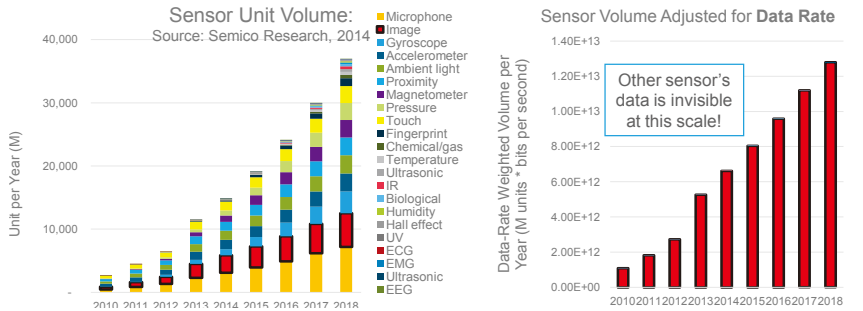
cadence

Real-Time Neural Networks
Automotive ADAS example



cadence

Vision Is the Computing Challenge
Growing data + compute drives new SoC designs



Cisco: "Consumer internet video traffic will be 80 percent of all consumer Internet traffic in 2019"
Source: Cisco May 2015: "Cisco Visual Networking Index: Forecast and Methodology, 2014-2019 White Paper"

cadence

CNN Complexity Has Been Growing Fast

- Today's deep learning industry motto is "Bigger Is Better"

Network	Application	Layers	GMACs
LeNet-5 for MNIST (1998)	Handwritten digit recognition	7	0.08
AlexNet (2012)	Imagenet	8	1.1
Deepface (2014)	Face recognition	8	1.4
FaceNet (2014)	Face recognition	22	1.6
VGG-19 (2015)	Imagenet	19	19.6
ResNet (2015)	Imagenet	152	11.3

- But, for **embedded devices**, power, price, and form-factor requirements cannot accommodate this trend

Courtesy of Dr. Stephen Hicks, Nuffield Department of Clinical Neurosciences, University of Oxford

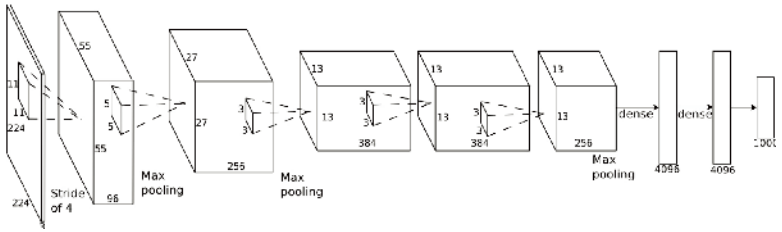
cadence

Neural Networks Efficiency

Need too much memory bandwidth ➔
Need too much compute ➔
Not accurate enough ➔

Example: AlexNet

- ~60M model parameters (FP32: 240MB)
- ~725M multiply accumulates (MACs) per image
- At 1000 images/sec: 240GB/s DDR bandwidth (FP32)
- Consider both compute efficiency and memory efficiency



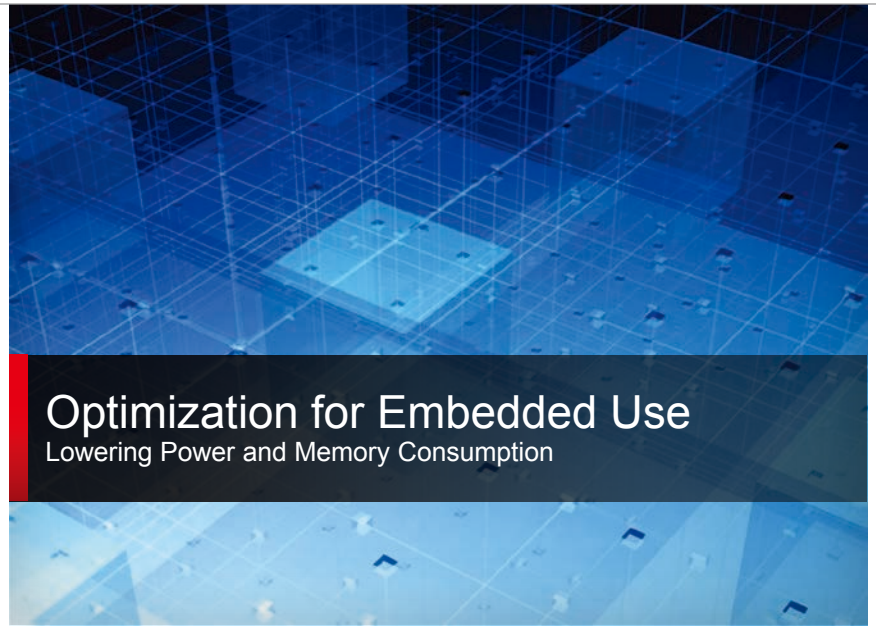
Can Neural Networks Map to Embedded Devices?

Why Neural Networks Are GOOD for Embedded Systems

- Highly flexible and adaptable to new tasks and new data
- Good for complex, noisy data from sensor-rich systems, e.g., imaging
- Deterministic throughput and latency
- Capable of achieving high efficiency with the right hardware and software approaches

Why Neural Networks May Be a PROBLEM for Embedded Systems

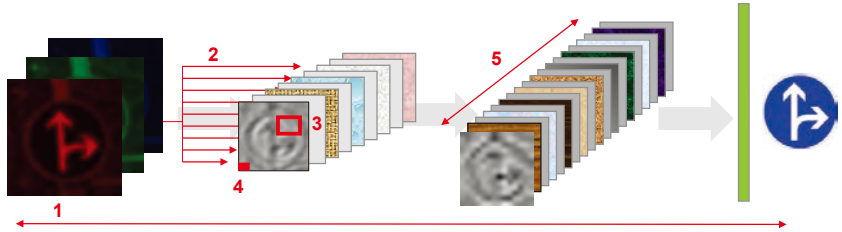
- Very compute intensive ➔ high power
- Large memory footprint ➔ high cost
- **Neural Networks**: Not well understood by embedded architects:
  - How it works and fits into the total data flow
  - How to test and measure performance
  - How to build good labeled datasets and train
- **Embedded Systems**: Not well understood by neural network experts
  - The significance of cost, power, bandwidth
  - The time-scales and costs for silicon integration



Optimization for Embedded Use

Lowering Power and Memory Consumption

Hyper Parameter Optimization



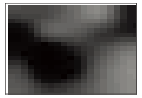
Network structure parameters with biggest impact on compute level:

- Number of layers
- Connectivity between layers
- Dimensions of convolution kernel per layer (n x n x depth)
- Data-types for weights and data: 32b float, 16b fixed, 8b fixed
- Number of feature maps per layer

We can optimize these to reduce CNN complexity

\+ Compute efficiency of target platform

Redundancies in Convolution Filter Weight Parameters



We can exploit these redundancies to reduce CNN complexity

  cadence

Methodology

- **Step 1**: Train a network as normal, using favorite tools/language
  - e.g., Caffe, TensorFlow, MatConvNet, theano, CNTK, torch, etc.


TensorFlow CNTK theano torch

- **Step 2**: Iteratively reduce hyper and weight parameters
  - Utilize:
    - Statistics
    - Linear algebra
  - Use the validation set as guidance for convergence

  cadence

Example: German Traffic Sign Recognition Benchmark (GTSRB)

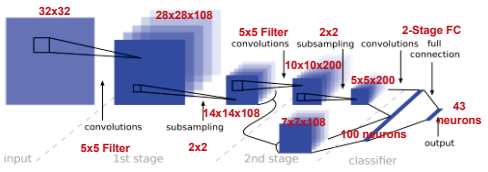
- 51840 images of German road signs in 43 classes
- Size of images varies between 15x15 to 222x193
- Images grouped by class and track with at least 30 images per track
- Images available as color images (RGB), HOG features, Haar features, and color histograms

  cadence

Menu of Algorithms

- TSR CNNs surpass human performance on the GTSRB benchmark
- Complexity of most recent, improved results has been growing at a fast pace

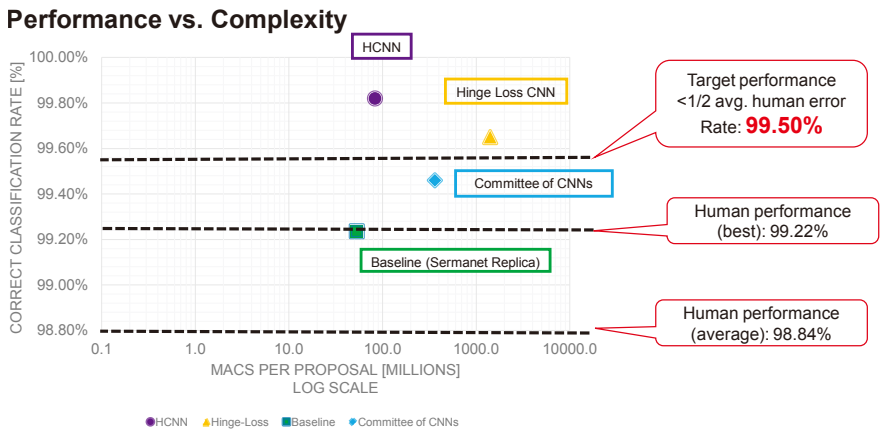
CCR%	Team	Method	Complexity [MMACs]
99.82	Cadence	Hierarchical CNN	83
99.65	Tsinghua	Hinge Loss Trained CNN	1409
99.46	IDSIA	Committee of CNNs	362
99.24	Cadence	Sermanet Replica	53
99.22	INI-RTCV	Human (best individual)	
99.17	Sermanet (NYU)	Updated multi-scale CNN	42
98.84	INI-RTCV	Human (average)	
98.31	Sermanet (NYU)	Multi-scale CNN	
96.14	CAOR	Random Forest	
95.68	INI-RTCV	LDA (HOG 2)	



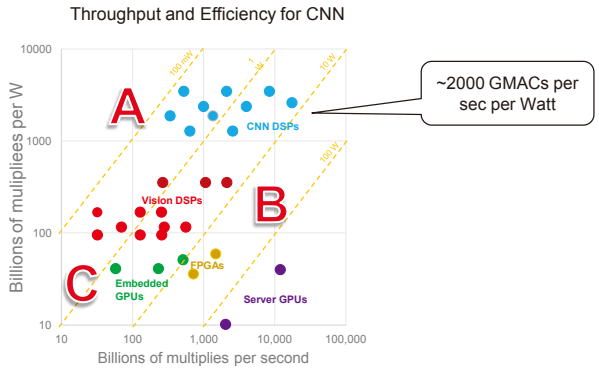
Source: Pierre Sermanet and Yann LeCun, "Traffic Sign Recognition with Multi-Scale Convolutional Networks", *IEEE IJCNN*, 2011

  cadence

Network Complexity Optimization in Performance-Complexity Space



Menu of Implementation Platforms



Typical ADAS Design Parameters

"The EyeQ4® ...[uses]…highly optimized architectures to support extremely intensive computations at automotive compliant ***power consumption of 2-3 Watts****.*

…'super-computer' ***capabilities of more than 2.5 teraflops*** *within a low-power (approximately 3W) automotive grade system-on-chip.*

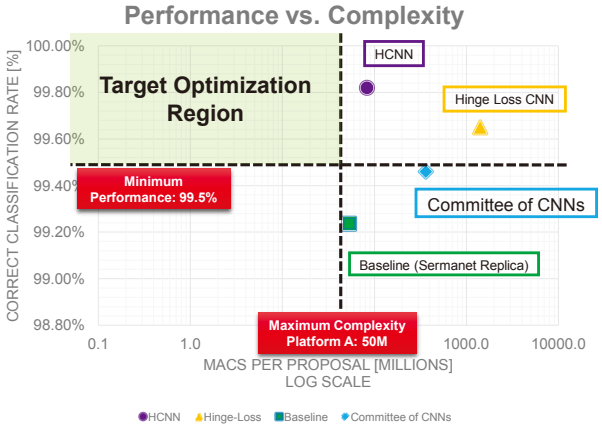
…cutting-edge computer vision algorithms like Deep Layered Networks and Graphical Models while ***processing information from 8 cameras simultaneously at 36 frames per second****."*

- Mobileye on PR Newswire, 2015

Mock Design Specifications

Parameter	Specification		
Performance: Correct Classification Rate	**99.50%** <1/2 avg. human error rate Superior to best human performance		
ROI Traffic Sign Proposal Average Rate	5,000 proposals per second		
Power Budget	0.125W (= 2W / 8 cameras x 50% budget)		
Platform	**A**	**B**	**C**
Efficiency [Peak GMACs per second per Watt]	2000	200	50
Maximum Throughput [GMACs per second]	250	25	6.25
Maximum MACs for CNN inference per proposal [Millions of MACs]	**50**	**5**	**1.25**

Network Complexity Optimization in Performance-Complexity Space



Goal: find a CNN that jointly satisfies Minimum Performance requirement and Maximum Computational Complexity allowed by the platform and the design parameters

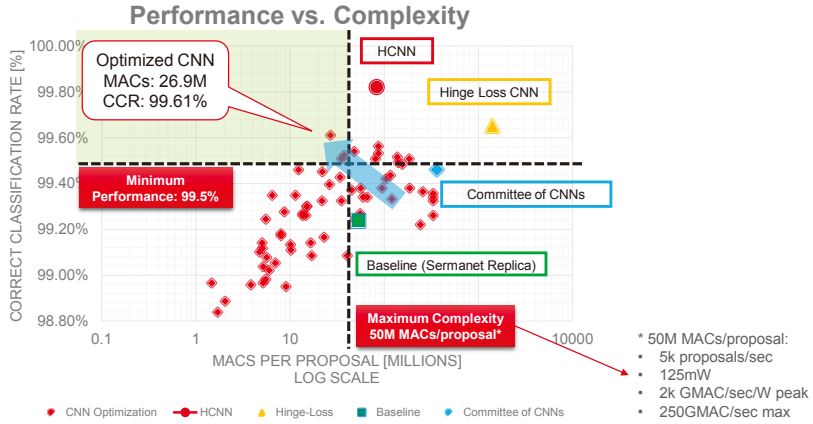
cadence

Samples of Iterative Network Optimization Process

Iteration	MACs [M]	CCR [%]	
1	366	99.26	
2	76	99.34	
3	21	99.45	
4	5	98.64	

cadence

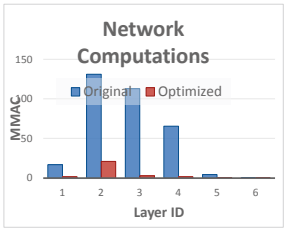
Network Complexity Optimization for Traffic Sign Recognition

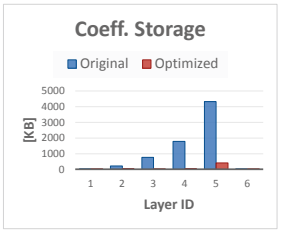


* 50M MACs/proposal:
- 5k proposals/sec
- 125mW
- 2k GMAC/sec/W peak
- 250GMAC/sec max

cadence

Analysis of Network Optimization for GTSRB





	Original	Optimized
Computations [MMAC]	331	26.9

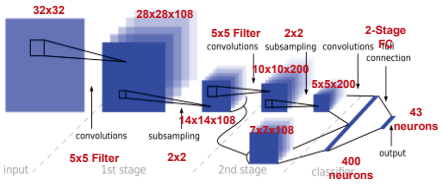
	Original 32b	Original 8b	Optimized 8b
Total Storage [Data+Weight]	28.8 MB	7.2 MB	0.54 MB

About 12X savings in computation,13-52X savings in memory

cadence

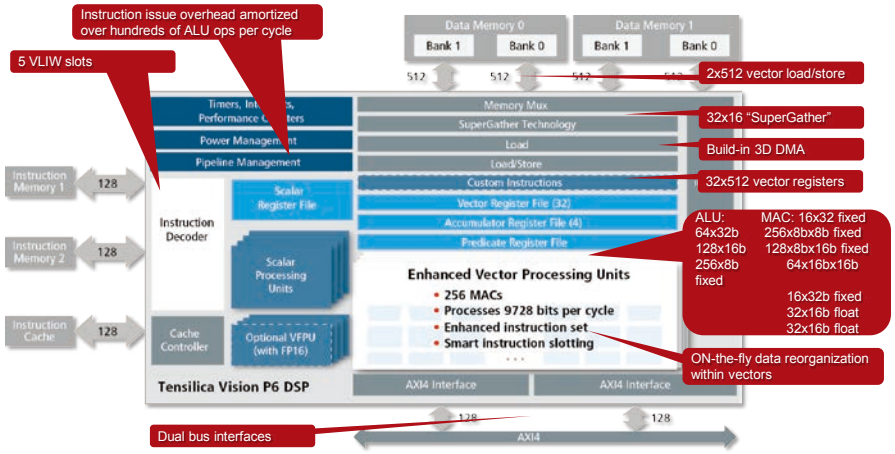
Limited and Mixed Precision Optimization

Conv. Layer 2 Precisions Data x Coeff	Correct Classification Rate	Complexity Reduction Factor
32b x 32b float	99.54%	1
8b x 8b	99.46%	10x
8b x 4b	99.43%	17x
4b x 4b	99.38%	26x



8b x 8b
all other layers

Tensilica Vision P6 DSP

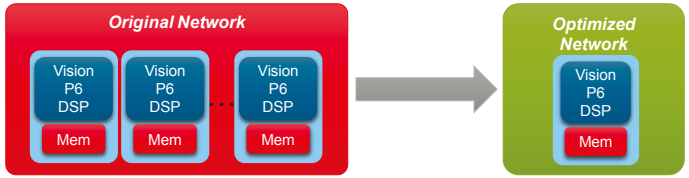


Vision P6 DSP: Overview

1.1GHz (16nm FF)	• Deeply pipelined design • Low-power clock gating
Highest per-cycle processing	• 4 vector ops per cycle, each 64-way SIMD • Processes 9728 bits per cycle • 256 ALU ops/cycle
Designed for embedded neural network applications	• 256 MACs/cycle • Optional: IEEE 32-bit vector floating-point with FP16
High data throughput	• Widest memory interface at 1024 bits • With SuperGather™ technology

Analysis of Network Optimization for GTSRB

System parameters with Vision P6 DSP



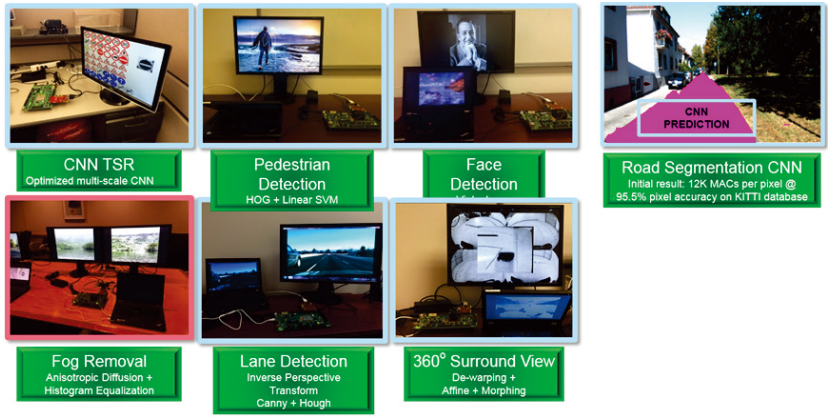
Supporting 5000 ROI/second			
Parameters	Original	Optimized	**Improvement**
# of Cores	12	1	12x
Area			12x
Computation	331 MMAC	27 MMAC	12x
DDR Size	7.2 MB/ROI	0.54 MB/ROI	13x
DDR Throughput	7.9 GB/s	1.0 GB/s	8x

cadence

Analysis of Network Optimization for GTSRB

System parameters with Vision P6 DSP

Cadence® example ADAS applications: In-house and in partnership



cadence

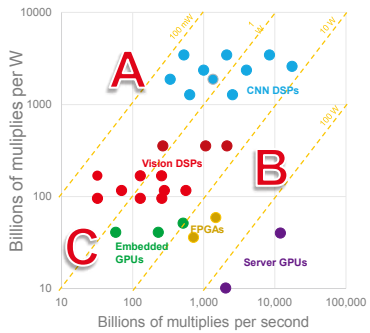
In Summary

cadence

Going Embedded with CNN

- Neural networks need lots of compute
  - Especially multiply-add
- Two key metrics
  - Scalability to high total compute
  - High multiply-add performance per watt
- Tensilica® Vision® DSPs often achieve greater efficiency than GPUs
- Clusters of cores are essential to performance scaling
- Addressing the memory bandwidth challenge:
  - Success with 16b, 8b, and even selective 4b representations
  - On-the-fly data compression/decompression exploits sparsity
  - **Together**: Up to 50X memory footprint reduction for coefficients is possible

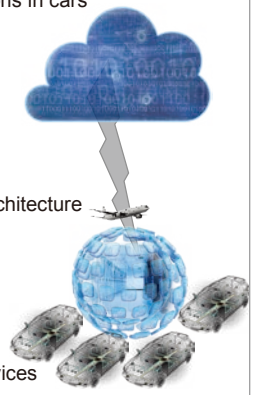
Throughput and Efficiency for CNN



cadence

The Road Ahead for Neural-Network-Based ADAS

- Neural networks will expand rapidly into **real-time embedded** functions in cars
- **Power** constraints and extreme **throughput** needs will drive CNN optimization in processor platforms—embedded and server
- Real-time neural network evolves from **object** recognition to **action** recognition
- Expect a **mad**—sometimes unguided—**scramble** for expertise, data, and applications
- >100X energy and >20X in bandwidth from **network AND engine** architecture optimization near-term
- In time: 1000 tera-MAC (**peta-MAC**) embedded neural networks
- Network optimization evolves from ad hoc exploration to automated "synthesis"**—a new kind of EDA**
- **New value chains** emerge—and swing between vertical integration and disintegration for new kinds of IP, tools, and data services
- **The player with the most road data wins—**Access to large training sets and massive simulation for algorithms gates progress to SAE Stage 5
- Potential **backlash** over privacy and "**rise of the machines**"



Find Out More From Cadence and Our Vision Partner Ecosystem

cadence®

NOTE